

SYSTEM AND METHOD FOR LEVERAGING NETWORKED COMPUTERS TO
VIEW WINDOWS BASED FILES ON LINUX PLATFORMS

FIELD OF THE INVENTION

5 [0001] The present invention relates generally to networked computing, and more specifically to, leveraging resources on a computer network to view a file on a client computer that is written in a format unreadable by the client computer.

BACKGROUND OF THE INVENTION

10 [0002] Thin client computers have recently gained in popularity for providing web browsing and electronic mail applications to users at low cost. The thin client computers achieve low cost by requiring less processing capabilities and memory space than standard PCs. In addition, the thin client computers are generally easier to maintain than standard PCs.

15 [0003] A popular operating system for managing the operation of thin client computers is a Linux operating system. Linux is popular because it may be obtained with little or no license fee. In addition, Linux is an open source operating system, allowing a programmer to easily configure it for the limited processing capabilities and memory space of many thin client computers.

20 [0004] A drawback of a Linux based computer, however, is that a user on the Linux based computer may have a difficult time viewing a file written in a Windows based format, such as Microsoft WordTM. Typically, the user on the Linux based computer has two options for viewing the file. The user may utilize a Linux based conversion utility to convert the file into an HTML file, which can then be viewed on the Linux based computer using a web browser application. However, Linux based conversion utilities generally do a moderately poor job of converting a Windows based file into an HTML file. Alternatively, the user may forward the file to a Windows based computer, and then view the file on the Windows based computer. However, this option may inconvenience the user by requiring the user to physically move to the Windows based computer to view

25 the file.

30

SUMMARY OF THE INVENTION

[0005] Embodiments of the present invention provide a system and a method for leveraging resources on a computer network to view a file on a client computer that is written in a format unreadable by the client computer.

5 [0006] In one aspect of a preferred embodiment, the client computer receives a file in a source format unreadable by the client computer. The client computer broadcasts the format of the received file to a plurality of File Viewing Servers (FVSs) in a computer network. In response, the client computer receives from at least one of the plurality of FVSs a conversion rating, which indicates the respective FVS's ability to convert the 10 source format of the received file unreadable by the client computer into a conversion format readable by the client computer. Upon receiving the conversion ratings from the plurality of FVSs, the client computer selects the FVS transmitting the highest conversion rating.

15 [0007] In another aspect of a preferred embodiment, the client computer stores a lookup table having one or more entries. Each entry includes a file format unreadable by the client computer and a selected one of the plurality of FVSs capable of converting the unreadable file format into a file format readable by the client computer. Upon receiving a file in a source format unreadable by the client computer, the client computer locates the entry in the lookup table corresponding to the source format of the received file. The 20 client computer then selects the FVS specified in the located entry.

25 [0008] In yet another aspect of a preferred embodiment, the computer network further includes a resource locator server. Upon receiving a file in a source format unreadable by the client computer, the client computer transmits the source format of the received file to the resource locator server. In response, the client computer receives a selection of one of the plurality of FVSs from the resource locator server. The selected server has the highest conversion rating corresponding to the source format of the received file.

30 [0009] To convert the source format of the received file into the conversion format, the client computer transmits the file to the selected FVS. The selected FVS uses a native application, *i.e.* an application residing locally on the selected FVS, to convert the received file written in the source format to the received file written in the conversion

format. The selected FVS then transmits the file to the client computer. Upon receiving the file, the client computer displays the file to a user.

[0010] In another aspect of a preferred embodiment, the selected FVS stores the received file written in the conversion format on its memory and transmits the location of the

5 stored file to the client computer. The client computer is then able to access the location of the file, and download the file written in the conversion format from the selected FVS onto the client computer.

[0011] In yet another aspect of a preferred embodiment, the client computer transmits the received file to the selected FVS. Upon receiving the file, the selected FVS opens the file

10 in a native application, *i.e.* an application residing locally on the selected FVS, and generates display content information therefrom. The selected FVS transmits the display content information to the client computer, and upon receipt, the client computer displays the display content information to a user.

[0012] Other methods, features and advantages of the invention will be or will become 15 apparent to one with skill in the art upon examination of the following figures and detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The drawings illustrate the design and utility of preferred embodiments of the 20 present invention, in which similar elements are referred to with common reference numerals.

[0014] FIG. 1 is a block diagram of an exemplary computer network according to an embodiment of the present invention.

[0015] FIG. 2 is a flow chart of steps for selecting a file viewing server on the computer 25 network.

[0016] FIG. 3 is a flow chart of steps for writing an entry into a file viewing server table.

[0017] FIG. 4 is a block diagram of an exemplary computer network according to another embodiment of the present invention.

[0018] FIG. 5 is a flow chart of steps for converting the format of a file using the 30 computer network.

[0019] FIG. 6 is a flow chart of steps for viewing a file on the computer network using a terminal emulator.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

5 [0020] FIG. 1 shows an exemplary computer network 5 according to one embodiment of the present invention. The computer network 5 includes a client computer 10 connectable to the Internet 15, e.g. the World Wide Web, via an Internet link 12, which may include, but is not limited to, an Ethernet link, a dialup link, a cable link, and the like. The computer network 5 further includes a plurality of File Viewing Servers (FVSs) 10 20a-20c, each FVS 20a-20c being connectable to the client computer 10 via a network connection 17a-17c. Although three FVSs 20a-20c are shown in FIG. 1, those skilled in the art will appreciate that any number of FVSs may be included in the computer network 5.

15 [0021] Each FVS 20a-20c and the client computer 10 may be part of the same local network and may communicate with each other through a local network connection 17a-17c, which may include, but is not limited to, an Ethernet connection, a Home Phone Network Adapter (HPNA) connection, and the like. Alternatively, each FVS 20a-20c may be located on a remote server on the Internet that is connectable to the client computer 10 via an Internet link 17a-17c.

20 [0022] The client computer 10 may be, but is not limited to, a thin client, a desktop computer, a laptop, a Wireless Access Protocol (WAP) mobile phone, a Personal Digital Assistant, or the like. The client computer 10 includes memory (not shown) for storing programs and data, such as RAM, EPROM, Flash memory, a hard drive, and the like. The client computer 10 also includes an operating system stored in memory for managing 25 the operation of the client computer 10. The operating system may be Linux, Microsoft WindowsTM, Mac OSTM, Unix, or any other computer operating system.

30 [0023] The client computer 10 further includes a web browser application stored in memory for downloading web pages onto the client computer 10 from the Internet 15 via an Internet link. The web browser application may be Netscape Navigator, Microsoft Internet ExplorerTM, an American Online (AOL) web browser, or any other web browser application. The client computer 10 may also include an e-mail application stored in

memory for receiving, writing and sending e-mail messages via the Internet link 12. The client computer 10 further includes a file viewing program module, such as a plug-in module, stored in memory for enabling the client computer 10 to display a file from a web page or an e-mail attachment that is written in a format unreadable by the client

5 computer 10. To accomplish this, the file viewing program module utilizes resources on one of the plurality of FVSs 20a-20c to convert the format of the file into another format readable by the client computer 10. The file viewing program module includes any combination of software-based modules and/or hardware-based modules that may perform the functions described herein.

10 [0024] The client computer 10 is connected to a display 25 for displaying information to a user, such as a LCD display, a CRT display, or the like. The client computer 10 is also connected to a user input interface 30 for entering user commands to the client computer 10, which may include, but is not limited to, a keyboard, a mouse, a touch screen, a track ball, and the like.

15 [0025] Each FVS 20a-20c is a computer, e.g. a PC, including resources that are available to the client computer 10 via the network connection 17a-17c. The available resources on each FVS 20a-20c include at least one native application on the respective FVS 20a-20c that is capable of converting the format of a file unreadable by the client computer 10 into another format readable by the client computer 10. For example, the native application may be a version of Microsoft WordTM that can convert a Word file into an HTML file, which is readable by the web browser application on the client computer 10.

20 [0026] Each FVS 20a-20c is assigned a conversion rating for each file format unreadable by the client computer 10 that the respective FVS 20a-20c is capable of converting into another file format readable by the client computer 10. The conversion rating indicates the ability of the respective FVS 20a-20c to convert a particular file format unreadable by the client computer into another file format readable by the client computer 10. The conversion rating may depend on several factors including, but not limited to, the native application used by the respective FVS 20a-20c for converting the unreadable file format into a readable file format and/or the processing capabilities of the respective FVS 20a-20c. For example, a FVS 20a-20c running a later version of Microsoft Word may be

assigned a higher conversion rating for converting Word files into HTML files than a FVS 20a-20c running an earlier version of Microsoft Word.

[0027] Steps for selecting one of the FVSs 20a-20c to convert the format of a file received by the client computer 10 will now be described with reference to FIG. 2. In 5 step 210, a user on the client computer 10 wishes to view a file from a web page or an e-mail attachment that is written in a format unreadable by the client computer 10. In other words, the file is written in a format that can not be read by a native application on the client computer 10. In step 220, the user indicates to the web browser application or the e-mail application on the client computer 10 that he or she wishes to view the file. For 10 example, the user may select a link to the file with the user input interface 30, such as by double clicking on the link with a mouse.

[0028] In step 230, the web browser or the e-mail application identifies the format of the file. In step 240, the web browser or e-mail application launches the file viewing 15 program module on the client computer 10 if the identified format is not readable by the client computer 10. The web browser application or the e-mail application may identify the format of the file by reading a Multipurpose Internet Mail Extension (MIME) attached to the file. The MIME is a standard identifier, typically a text string, identifying the format of a file. The web browser application or the e-mail application may include a table that links the MIME of a file with the file viewing program module so that the web 20 browser application or the e-mail application can automatically launch the file viewing program module when it reads the MIME of the file. In step 250, the file viewing program module broadcasts the format of the received file to the plurality of FVSs 20a-20c via the computer network connections 17a-17c. The file viewing program module may also identify the format of the file by reading the MIME attached to the file.

[0029] In step 260, each FVS 20a-20c that is capable of converting the broadcasted file 25 format unreadable by the client computer into another file format readable by the client computer 10 transmits its conversion rating for the broadcasted file format to the client computer 10. Preferably, each FVS 20a-20c transmits contact information along with its conversion rating to the client computer 10. The contact information may include, but is 30 not limited to, a Uniform Resource Locator (URL) and/or an Internet Protocol (IP) address of the respective FVS 20a-20c. Upon receiving the conversion ratings from the

FVSs 20a-20c, the file viewing program module selects the FVS 20a-20c transmitting the highest conversion rating in step 270. The file viewing program module may then contact the selected FVS 20a-20c using the contact information for the selected FVS 20a-20c to convert the format of the received file into another format readable by the client computer 10. Steps for carrying out the file conversion on the selected FVS 20a-20c will be discussed later.

5 [0030] In some cases, the FVSs 20a-20c may not know in advance which file format is readable by the client computer 10. In these cases, the client computer 10 may broadcast both the format of the received file and a conversion format to the FVSs 20a-20c, where 10 the conversion format is a format readable by the client computer 10. In response, each FVS 20a-20c transmits its conversion rating corresponding to the format of the received file and the conversion format to the client computer 10.

10 [0031] In another embodiment of the present invention, the file viewing program module uses a FVS lookup table stored on the memory of the client computer 10 to determine 15 which FVS 20a-20c is most suitable for converting the format of a received file unreadable by the client computer into another format readable by the client computer 10. In this embodiment, the FVS lookup table lists various file formats and a preferred FVS 20a-20c for each file format. The FVS lookup table may further list contact information for each listed FVS 20a-20c including, but not limited to, a URL and/or an IP address for 20 the respective FVS 20a-20c.

25 [0032] When the user wishes to view a file written in a format that is not readable by the client computer 10, the file viewing program module searches for the format of the file in the FVS lookup table. If the file viewing program module is able to locate the format in the FVS lookup table, then the file viewing program module contacts the corresponding FVS 20a-20c in the FVS lookup table using the contact information provided by the FVS 30 lookup table.

[0033] FIG. 3 shows exemplary steps for writing an entry into the FVS lookup table. The entry includes a file format unreadable by the client computer 10 and the FVS 20a-20c most suitable for converting the unreadable file format into another file format readable by the client computer 10. In step 310, the file viewing program module broadcasts the unreadable file format to the plurality of FVSs 20a-20c via the computer

network connections 17a-17c. In step 320, each FVS 20a-20c capable of converting the broadcasted file format into another file format readable by the client computer 10 transmits its conversion rating for the broadcasted file format to the client computer 10. Upon receiving the conversion ratings from the FVSs 20a-20c, the file viewing program module selects the FVS 20a-20c transmitting the highest conversion rating in step 330.

5 In step 340, the file viewing program module enters the file format and the selected FVS 20a-20c into the FVS lookup table. The file viewing program module may repeat steps 310 through 340 for each one of several different file formats, so that each file format and the corresponding selected FVS 20a-20c is entered into the FVS lookup table.

10 [0034] The FVS lookup table may include more than one FVS 20a-20c for a file format. For example, the FVS lookup table may also include the FVS 20a-20c having the second highest conversion rating for a file format in case the FVS 20a-20c having the highest conversion rating for the file format is disconnected from the computer network 5 or otherwise becomes unavailable.

15 [0035] FIG. 4 shows a computer network 410 according to another embodiment of the invention. The computer network 410 includes a resource locator server 420, which is connectable to the client computer 10 via a network connection 430 and the FVSs 20a-20c via network connections (not shown). In this embodiment, when the user wishes to view a file written in a format unreadable by the client computer 10, the file viewing

20 program module on the client computer 10 transmits the format of the file to the resource locator server 420. Upon receiving the format from the client computer 10, the resource locator server 420 selects the FVS 20a-20c that is most suitable for converting the received format into another format readable by the client computer 10. The resource locator server 420 transmits the selected FVS 20a-20c, along with contact information for

25 the selected FVS 20a-20c, to the client computer 10. The contact information may include, but is not limited to, a URL and/or an IP address of the selected FVS 20a-20c. The file viewing program module then contacts the FVS 20a-20c selected by the resource locator server 420 to convert the format of the received file into another format readable by the client computer 10.

30 [0036] The resource locator server 420 may select the FVS 20a-20c for a received format by following steps 250-270 described above. In this case, the resource locator server 420

broadcasts the received format to the plurality of FVSs 20a-20c. In response, each FVS 20a-20c transmits its conversion rating for the broadcasted format to the resource locator server 420. The resource locator server 420 then selects the FVS 20a-20c transmitting the highest conversion rating. In addition to using the conversion ratings in selecting a

5 FVS 20a-20c, the resource locator server 420 may consider other factors in selecting the FVS 20a-20c. For example, the resource locator server 420 may keep track of the current workloads of the FVSs 20a-20c and give preference to a FVS 20a-20c having a smaller workload over a FVS 20a-20c having a larger workload. This will tend to balance the workload among the FVSs 20a-20c on the computer network 410.

10 [0037] Alternatively, the resource locator server 420 may use a FVS lookup table to select the FVS 20a-20c. The resource locator server 420 may write entries for various file formats into its FVS lookup table by following steps 310-340 described above for each file format. In addition, the resource locator server 420 may update the entries in its FVS lookup table each time one or more FVSs 20a-20c are added to or taken away from

15 the computer network 410.

[0038] Steps for contacting the selected FVS 20a-20c to convert a file written in a format unreadable by the client computer 10 will now be described with reference to FIG. 5. In step 510, the file viewing program module transmits the received file to the selected FVS 20a-20c via the network connection 17a-17c. In step 520, the selected FVS 20a-20c

20 receives the file and stores the file locally in its memory. In step 530, the selected FVS 20a-20c instructs a native application to convert the format of the received file into another format readable by the client computer 10. In step 540, the FVS 20a-20c stores the converted file in its memory. For example, the selected FVS 20a-20c may convert a Word file into an HTML file by instructing a version of Microsoft WordTM to open and then save the Word file as an HTML file to a specified location in the FVS's 20a-20c memory.

25 [0039] In step 550, the FVS 20a-20c transmits the converted file to the client computer 10 via the network connection 17a-17c. In step 560, the file viewing program module displays the received converted file on the display 25 using a native application on the client computer 10. Alternately, in step 550, the FVS 20a-20c may transmit the location of the converted file in its memory to the client computer 10. The location may be

transmitted to the client computer 10 in the form of a URL and/or an IP address. In step 560, the file viewing program module directs the web browser application on the client computer 10 to the location of the converted file on the FVS 20a-20c so that the web browser application can download and display the converted file to the user.

5 [0040] An advantage of the present invention is that it enables a Linux based client computer 10 to view a Windows based file, such as a Word, Power Point, Visio, BMP, Excel, and the like file. This is accomplished by having the file viewing program module transmit the Windows based file to a Windows based FVS 20a-20c to convert the received Windows based file into an HTML file, which is readable by the web browser 10 application on the client computer 10.

[0041] In another embodiment, any one of the computer networks 5, 410 may include a terminal emulator, such as Citrix and Virtual Network Computing (VNC). The terminal emulator enables applications running on one of the FVSs 20a-20c to be displayed on the client computer 10. The terminal emulator includes a server application on the respective 15 FVSs 20a-20c and a viewer application on the client computer 10. The server application on the respective FVS 20a-20c converts an application running on the FVS 20a-20c into display content information, which is transmitted to the client computer 10 via the network connection 17a-17c. The viewer application on the client computer 10 uses the received display content information to display the application running on the FVS 20a-20c in a terminal window on the display 25 of the client computer 10. Many 20 commercially available terminal emulators, such as VNC, allow the server application and the viewer application to run on different operating systems and/or computer architectures.

[0042] Steps for viewing a file on the computer network using the terminal emulator will 25 now be described with reference to FIG. 6. In step 610, the file viewing program module on the client computer 10 transmits the received file to one of the FVSs 20a-20c via the network connection 17a-17c. In step 620, the FVS 20a-20c receives and stores the file in its memory. In step 630, the FVS 20a-20c loads and runs a native application on the FVS 20a-20c, which is capable of reading the received file. In step 640, the FVS 20a-20c 30 instructs the native application running on the FVS 20a-20c to open the received file. For example, the FVS 20a-20c may instruct Microsoft Word to open a Word file. In step

650, the server application on the FVS 20a-20c converts the application running on the FVS 20a-20c into display content information. In step 660, the server application transmits the display content information to the client computer 10 via the network connection 17a-17c. In step 670, the viewer application on the client computer 10 uses 5 the received display content information to display the application running on the FVS 20a-20c in a terminal window on the display 25 of the client computer 10. This enables the user on the client computer 10 to view the file, which is opened using the application running on the FVS 20a-20c.

[0043] The conversion of the format of a file into another format may also include 10 decrypting an encrypted file or decompressing a compressed file, such as a ZIP file. This may be useful in cases where the client computer 10 does not have a native application capable of decrypting an encrypted file or decompressing a compressed file. In these cases, the file viewing program module may transmit an encrypted or compressed file to the one of the FVSs 20a-20c. Upon receiving the file, the respective FVS 20a-20c 15 decrypts or decompresses the file and stores the decrypted or decompressed file in its memory. The FVS 20c may then make the decrypted or decompressed file available to the client computer 10 by transmitting the decrypted or decompressed file to the client computer 10 or transmitting the location of the decrypted or decompressed file in the FVS's 20a-20c memory to the client computer 10. For cases in which the decrypted or 20 decompressed file is written in a format that is still not readable by the client computer 10, the FVS 20a-20c may further convert the format of the decrypted or decompressed file into another format that is readable by the client computer 10.

[0044] In another embodiment, the file viewing program module on the client computer 10 may transmit a compressed file archive containing several files to the selected FVSs 25 20a-20c. In this embodiment, the respective FVS 20a-20c decompresses the received file archive and stores the decompressed file archive in its memory. The FVS 20a-20c generates an HTML index page listing each decompressed file contained in the decompressed file archive along with a link to each decompressed file, and stores the 30 HTML index page in the FVS's 20a-20c memory. The FVS 20a-20c then transmits the HTML index page to the client computer 10, and the file viewing program module displays the HTML index page to the user using the web browser application on the

client computer 10. Alternatively, the FVS 20a-20c transmits a location, *e.g.* a URL, of the HTML index page to the client computer 10, and the file viewing program module then directs the web browser application on the client computer 10 to the location of the HTML index page so that the web browser can download and display the HTML index

5 page to the user. From the HTML index page, the user can view any one of the decompressed files listed in the HTML index page by clicking onto the file's link on the HTML index page. This embodiment may be especially useful in cases where the client computer 10 lacks enough available memory space to store the entire decompressed file archive at one time.

10 [0045] In a further embodiment, the selected FVS 20a-20c may download a file archive from the Internet 15 and decompress the file archive on behalf of the user on the client computer 10. In this embodiment, the user may click onto the link to a file archive on a web page that he or she wishes to view. The file viewing program module then transmits the URL of the file archive to the selected FVS 20a-20c along with a request to download

15 and decompress the file archive. The selected FVS 20a-20c uses the received URL to locate the file archive on the Internet 15. After locating the file archive on the Internet 15, the FVS 20a-20c downloads and decompresses the file archive and stores the decompressed file archive in its memory. The FVS 20a-20c may then make the decompressed file archive available to the user on the client computer 10 using the

20 HTML index page according to the previous embodiment.

[0046] Those skilled in the art will appreciate that various modifications may be made to the just described preferred embodiments without departing from the spirit and scope of the invention. For example, those skilled in the art will appreciate that the source code of the file viewing program module can be integrated into the web browser application on the client computer 10. Therefore, the invention is not to be restricted or limited except in accordance with the following claims and their legal equivalents.